Hierarchical Dialogue Optimization Using Semi-Markov Decision Processes

Heriberto Cuayáhuitl[1], *Steve Renals*[1], *Oliver Lemon*[2], *Hiroshi Shimodaira*[1]

CSTR[1], HCRC[2], School of Informatics, University of Edinburgh
2 Buccleuch Place, EH8 9LW, Edinburgh, Scotland, UK
{h.cuayahuitl,s.renals,olemon,h.shimodaira}@ed.ac.uk

Abstract

This paper addresses the problem of dialogue optimization on large search spaces. For such a purpose, in this paper we propose to learn dialogue strategies using multiple Semi-Markov Decision Processes and hierarchical reinforcement learning. This approach factorizes state variables and actions in order to learn a hierarchy of policies. Our experiments are based on a simulated flight booking dialogue system and compare flat versus hierarchical reinforcement learning. Experimental results show that the proposed approach produced a dramatic search space reduction (99.36%), and converged four orders of magnitude faster than flat reinforcement learning with a very small loss in optimality (on average 0.3 system turns). Results also report that the learnt policies outperformed a hand-crafted one under three different conditions of ASR confidence levels. This approach is appealing to dialogue optimization due to faster learning, reusable subsolutions, and scalability to larger problems.
Index Terms: Spoken dialogue systems, semi-Markov decision processes, hierarchical reinforcement learning.

1. Introduction

Spoken dialogue systems that learnt to optimize their behaviour have typically been investigated within the flat tabular reinforcement learning paradigm [1, 2, 3, 4, 5]. However, the scalability of this approach is limited due to the fact that search spaces grow exponentially according to the state variables taken into account (referred as "the curse of dimensionality"). Even systems with simple state representations may have large search spaces with quick growth towards intractability. Recent investigations employ function approximation [6] and prior knowledge [5] in order to find solutions on reduced search spaces. They have been applied to small-scale systems aiming for a single global solution. However, little attention has been devoted to finding solutions with the divide-and-conquer approach.

Previous work in the literature of artificial intelligence has investigated divide-and-conquer approaches to address the problem of reinforcement learning on large search spaces, which is broadly referred as Hierarchical Reinforcement Learning (HRL). The fundamental theory behind HRL is based on Semi-Markov Decision Processes (SMDPs) [7]. HRL is appealing due to the following benefits: a) improved exploration, because exploration can take multi-time steps by using low-level and high-level actions; b) reduced computational demands, because breaking a problem into subproblems helps to ignore irrelevant features of the flat environment state; and c) knowledge transfer, because solutions learnt on previous problems can be reused in new problems. However, the price to pay for such benefits is that HRL methods may learn *near* optimal solutions. Nevertheless, HRL methods learn the best policies according to the constraints specified in the hierarchy [8].

Related work on SMDPs and HRL can be broadly classified into two approaches: those that learn on a single SMDP and those that learn on multiple SMDPs. Firstly, methods learning on a single SMDP have focused on high-level and low-level actions in order to speed up learning [9][10]. Although this approach helps to reduce the curse of dimensionality problem, it is still limited because the environment is represented with flat states rather than hierarchical states. This means that learning using a single SMDP lacks scalability and knowledge transfer. Secondly, methods learning on multiple SMDPs can employ hierarchical states, actions and rewards. The use of hierarchical states is very useful in order to ignore irrelevant state variables, meaning that smaller solutions can be found faster, with reduced computational demands, and with opportunities to reuse policies [8]. Other related work concerns POMDPs for dialogue management under uncertainty; however, its application is still limited to small-scale dialogue systems because this approach is only feasible with small state-action spaces [4, 11].

In this paper we investigate learning dialogue strategies using multiple Semi-Markov Decision Processes and Hierarchical Reinforcement Learning. This approach has not been applied before to dialogue optimization (though see [12] for a different HRL approach), and we will show that it is a promising method to efficiently optimize the behaviour of large-scale spoken dialogue systems.

2. Semi-Markov Decision Processes

SMDPs are a generalization of MDPs, aiming to model complex decision making problems following the well known divide-and-conquer approach using either a single or multiple models. In the latter, the dividing part consists in breaking an MDP into a hierarchy of SMDPs (subtasks), where each SMDP will be referred to as a "composite action" lasting multiple time-steps, and each single time-step action will be referred to as a "primitive action". In contrast with MDPs having only primitive actions, SMDPs can have both types of actions. The conquering part consists in learning unified solutions for each subtask until finding a (near)optimal solution for the root subtask.

A discrete-time SMDP is a 4-tuple $M =< S, A, T, R >$ characterized as follows: S is a set of environment states; A is a set of actions; T is a transition function that observes the next state s' given the current state s and action a with probability $P(s', \tau|s, a)$; and $R(s, a)$ is the reward function that specifies the rewards given to the agent for choosing action a when the environment makes a transition from s to s'. The random variable τ denotes the number of time-steps that take to execute action a in state s [7][8]. An MDP can be decomposed into multiple SMDPs hierarchically organized in L layers and M models per layer, denoted as $\mathcal{M} = \{M_j^i\}$, where $j \in \{0, ..., |M| - 1\}$ and $i \in \{0, ..., |L| - 1\}$. In this way, any given SMDP in the

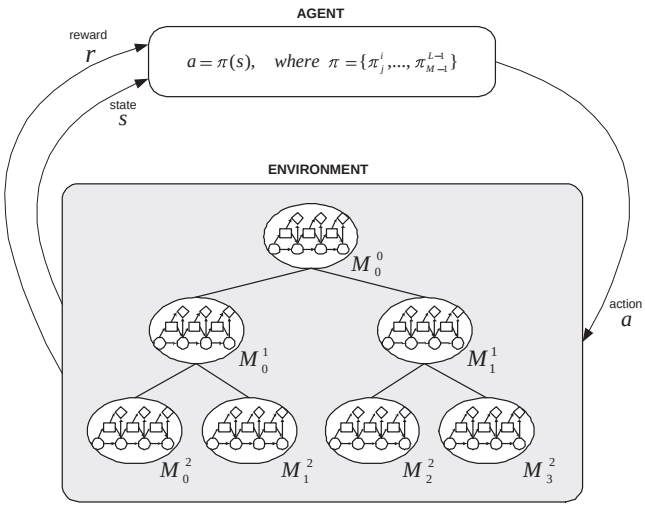

Figure 1: *Architecture of the agent-environment interaction for hierarchical reinforcement learning using multiple SMDPs.*

hierarchy is denoted as $M_j^i =< S_j^i, A_j^i, T_j^i, R_j^i >$, see figure 1.

The goal in an SMDP is to find a (near) optimal policy π^*, that maximizes the reward of each state. The optimal value function denoted as $V^*(s)$, is the expected cumulative reward of s under π^*. Similarly, the optimal action-value function denoted as $Q^*(s,a)$, is the expected cumulative reward for executing a in s following π^*. The Bellman equations for V^* and Q^* of subtask M_j^i can be expressed as

$$V_j^{*i}(s) = \max_a \left[R_j^i(s,a) + \sum_{s',\tau} \gamma^\tau P_j^i(s',\tau|s,a) V_j^{*i}(s') \right], \quad (1)$$

$$Q_j^{*i}(s,a) = R_j^i(s,a) + \sum_{s',\tau} \gamma^\tau P_j^i(s',\tau|s,a) \max_{a'} Q_j^{*i}(s',a'). \quad (2)$$

Finally, the optimal policy for each subtask is given by equation 3. This policy can be found by dynamic programming or RL algorithms for SMDPs, the latter are preferred [7].

$$\pi_j^{*i}(s) = \arg\max_a Q_j^{*i}(s,a), \forall s \in S_j^i \text{ and } a \in A_j^i. \quad (3)$$

2.1. Hierarchical Reinforcement Learning Algorithms

The goal of Hierarchical Reinforcement Learning (HRL) is to find solutions for complex Markov decision problems. To date, few methods have been investigated for learning a hierarchy of SMDPs such as HSMQ-Learning [13] and the MAXQ learning algorithms [8]. These hierarchical reinforcement learning algorithms share the following properties: a) They execute subtasks using the well known stack mechanism, b) they find solutions with a form called "recursive optimality" rather than "hierarchical optimality" or "global optimality" - making possible to find context-independent (reusable) policies, and c) they converge to recursively optimal policies with similar convergence properties as Q-Learning [14].

3. Hierarchical Dialogue Optimization

The idea of hierarchical dialogue optimization consists in finding a decision maker that takes the best high-level and low-level actions for every different situation in the conversation. In the rest of the paper we address the question "How to apply the theory of SMDPs and HRL to spoken dialogue systems?"

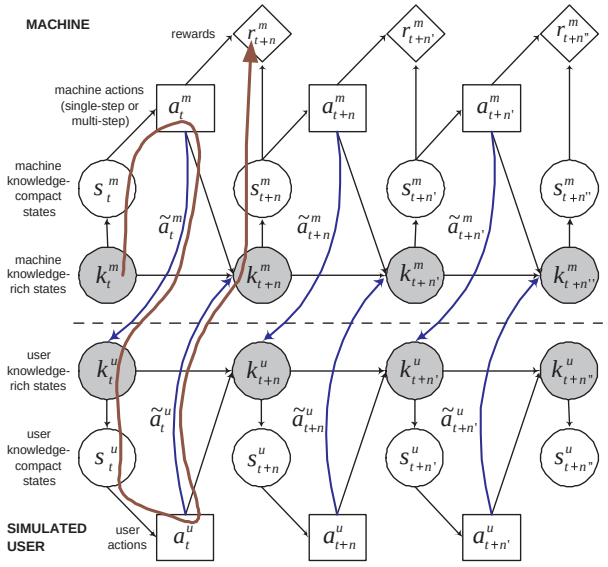

Figure 2: *An SMDP interacting with a simulated user when machine actions are dialogue acts a_t^m or $\tilde{a}_t^m$ (distorted). The long line shows a sample interaction, where state transitions are observed from states κ_t^m holding information about the dialogue.*

3.1. Dialogue as a Semi-Markov Decision Process (SMDP)

We propose treating the problem of dialogue optimization as a Semi-Markov Decision Process, which employs hierarchical dialogues rather than flat dialogues. Based on notation from figure 2, a hierarchical dialogue consists of a sequence of tuples $(s_1^m, a_1^m, s_1^u, a_1^u)$ and subdialogues D_t^τ lasting τ time-steps. Equations 4-5 denote a dialogue with a child subdialogue, hence a dialogue with decendants form a hierarchy of subdialogues, where every subdialogue is optimized with a separate SMDP.

This work assumes deterministic state transitions in the SMDPs in order to ensure coherent and consistent dialogues, see figure 2. Briefly, the conversants use knowledge-rich states κ^m and κ^u, which do not enumerate the vast combinations, they only keep the current state of the world. The knowledge-rich states κ_t employ ontological dialogue structures that represent knowledge about the conversation (e.g., dialogue goals, semantic frames, slots, retries, database, last conveyed dialogue acts[1], etc.) Finally, the knowledge-compact states s_t used to choose actions they do enumerate a compact number of combinations.

$$D_1^T = \{(s_1^m, a_1^m, s_1^u, a_1^u), ..., D_t^{t+\tau}, ..., (s_T^m, a_T^m, s_T^u, a_T^u)\} \quad (4)$$

$$D_t^{t+\tau} = \{(s_t^m, a_t^m, s_t^u, a_t^u), ..., (s_{t+\tau}^m, a_{t+\tau}^m, s_{t+\tau}^u, a_{t+\tau}^u)\} \quad (5)$$

3.2. Decomposing a Dialogue Manager into Subtasks

We propose a two-stage approach to divide a dialogue task into subtasks. The first stage decomposes a dialogue manager based on its set of dialogue goals. Let $G = \{g1, ..., g_X\}$ be the set of dialogue goals. This decomposition will have X subtasks. The second stage decomposes every subtask X into a set of slot filling strategies (e.g., mandatory slots, optional slots, terminal slot). Let $F = \{f_1, ..., f_Y\}$ be the set of slot filling strategies. In this way, the global decomposition will have $1+Y$ or $1+X+XY$ subtasks for a single-goal or multi-goal dialogue system,

[1]Sample dialogue acts for a single interaction of both conversants: a^m =request(date), a^u =provide(date=1Dec2007,time=morning).

Table 1: *State variables for the flight booking dialogue system.*

Variable	Values	Description
SL0	$\{0, 1, 2, 3, 4\}$	Status of slot "departure city"
SL1	$\{0, 1, 2, 3, 4\}$	Status of slot "destination city"
SL2	$\{0, 1, 2, 3, 4\}$	Status of slot "date"
SL3	$\{0, 1, 2, 3, 4\}$	Status of slot "time"
SL4	$\{0, 1, 2, 3, 4\}$	Status of slot "airline"
SL5	$\{0, 1, 2, 3, 4\}$	Status of slot "flight offer"
SIF	$\{0, ..., 5\}$	Slot in focus
DBT	$\{0, 1, 2\}$	Number of database tuples (1=none, 2=few, 3=many)

Table 2: *Action space for the flight booking dialogue system.*

Action	Description
req	Request slot in focus
apo+req	Apology for mis-recognition + request slot in focus
sic+req	Single implicit confirmation + request slot in focus
mic+req	Multiple implicit confirmation + request slot in focus
sec	Single explicit confirmation of the slot in focus
mec	Multiple explicit confirmation of filled slots
acc	Move to the next ascending slot with lower-value
dbq+sta	Perform a database query + inform its tuples size
pre+ofr	Information presentation + offer options
apo+ofr	Apology for mis-recognition + offer options

respectively. This simple heuristic aims to be a guideline for specifying the hierarchy of subtasks. Finding the best hierarchy for a given dialogue system is beyond the scope of this paper.

4. Experiments and Results

The aim of our experiments was to compare flat versus hierarchical Reinforcement Learning (RL) when flat learning is still feasible. Our experiments are based on a six-slot mixed-initiative flight booking spoken dialogue system employing a simulated conversational environment at the dialogue-act level.

4.1. Experimental Setup

For flat RL, the search space representation has 8 non-binary state variables[2] and 10 primitive actions (see tables 1 and 2). The reward function[3] focuses on efficient dialogues and is given by equation 6. For hierarchical RL, the hierarchical search space representation has 4 subtasks (one parent and three children) with $[4, 5, 1, 1]$ non-binary state variables[4] and $[4, 7, 3, 3]$ actions, respectively (see figure3). The reward function is the same as the one used for flat learning, used in each subtask.

Our experiments employed the simulated environment described in section 4.2. The learning setup used Q-Learning for flat learning [14] and HSMQ-Learning for hierarchical learning [13]. Finally, the learning parameters are the same for both flat and hierarchical learning: step size $\alpha = 100/(100 + t)$, with t elapsed time-steps; discount factor $\gamma = 1$; selection strategy using ϵ-Greedy, with $\epsilon = 0.01$; and initial Q-values of 0.

$$R = \begin{cases} 0 & \text{for successful dialogue (sub)goal} \\ -10 & \text{for presenting too much or no information} \\ -1 & \text{otherwise} \end{cases} \quad (6)$$

[2]Slot values: 0=unknown, 1=known with *low* confidence, 2=known with *medium* confidence, 3=know with *high* confidence, 4=confirmed.

[3]Illegal actions had no effect in the dialogues and only wasted time, e.g., request an already filled slot, request an already confirmed slot, etc.

[4]Values of state variables MAN, OPT, and TER in the root subtask (see figure 3): 0=unfilled slots, 1=filled slots, 2=confirmed slots.

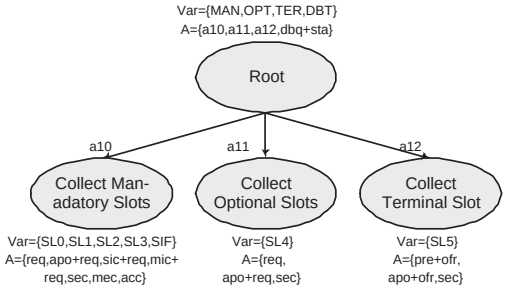

Figure 3: *A task hierarchy for the six-slot flight booking system, where each subtask is represented as a separate SMDP.*

4.2. The Simulated Environment

The simulated environment consisted of three components: user behaviour, ASR behaviour, and database behaviour. We used a hand-crafted semi-stochastic simulated user model with consistent user responses. Briefly, the simulated user behaved according to the following probabilities: probability of in-vocabulary words $p(\text{iv}) = 0.9$, probability of obedience $p(\text{ob}) = 0.9$ (for filling the slot in focus), probability of reproviding information in negative confirmations $p(\text{ri}) = 0.8$, and probability of filling multiple slots $p(\text{ms}) = 0.4$. The ASR model generated keyword substitutions with probability of $p(\text{sub}) = 0.2$ and the confidence levels were generated randomly with three different distributions for low, medium, and high values: a) $[1/4, 1/4, 1/2]$, b) $[1/3, 1/3, 1/3]$, and c) $[1/2, 1/4, 1/4]$. This setup aimed to test the learning agents and a baseline dialogue strategy under different conditions. The database queries produced the following outcomes: a large number of tuples after filling the first four slots, a small number of tuples after filling the first five slots, an empty set was observed otherwise.

In addition, in order to test the quality of the learnt policies, we hand-crafted a deterministic dialogue strategy for our case study. Its behaviour is briefly as follows: a) if slot in focus is unknown then collect information (with implicit confirmation if there were any filled slots), b) if slot in focus is known with low confidence then do an apology, c) if slot in focus is known with medium confidence then do an explicit confirmation, and d) if slot in focus is known with high confidence then move the slot in focus to the next ascending one with lower value. We used this dialogue strategy as baseline of system behaviour.

4.3. Results

Experimental results show that the hierarchical search space obtained a dramatic reduction of 99.36%. Table 3 shows the number of state-actions for both flat (2.8 million) and hierarchical (18K) search spaces. The learnt dialogue strategies generated coherent and consistent conversations. Figure 4 shows the learning curves of the dialogue policies, averaged over 10 training runs of 10^5 episodes. The three plots illustrate different amounts of ASR confidence levels. The first thing to notice is that hierarchical RL learns faster than flat RL by four orders of magnitude. The second thing to notice is that the deterministic strategy performs almost as well as the learnt policies for only one situation, but in general it was outperformed by the learnt policies. This illustrates the benefits of using dialogue optimization where more efficient conversations can be achieved by using (near) optimal dialogue strategies. However, this result also suggests that the simulated environment must reflect as much

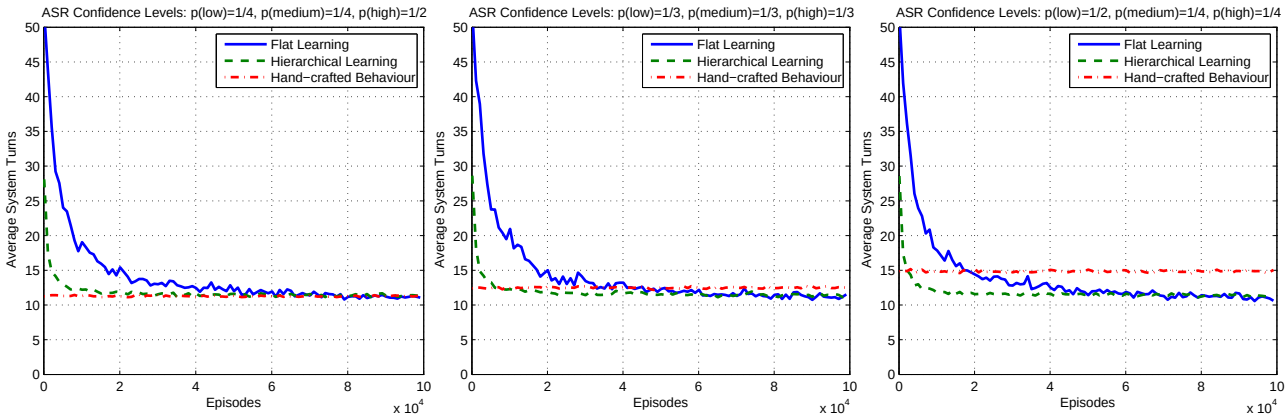

Figure 4: *Learning curves of policies using flat and hierarchical reinforcement learning in the 6-slot flight booking dialogue system. The best learnt policy outperformed the hand-crafted one by* 0.2, 1.3, *and* 3.7 *system turns on average in all cases (from left to right).*

Table 3: *Search spaces for flat and hierarchical learning.*

Learning Approach	States	Actions	$\|S \times A\|$
Flat	281250	10	2812500
Hierarchical	[81,2500,5,5]	[4,7,3,3]	17854

as possible the behaviour of the real environment, otherwise the learnt dialogue policies will no longer be optimal. The last thing to notice is that flat learning eventually performs slightly better than hierarchical learning. An evaluation on the last 10^4 episodes (dialogues) reports that flat learning achieves slightly more efficient conversations, on average 0.3 system turns less than hierarchical learning (at $p < 0.01$). Presumably because in the hierarchical setting the optional slot ("airline") cannot be explicitly confirmed together with the mandatory slots. Nevertheless, for practical purposes this loss in optimality may be well worth the gains in terms of scalability to larger problems.

5. Conclusions and Future Work

In this paper we proposed to learn dialogue strategies using multiple Semi-Markov Decision Processes and Hierarchical Reinforcement Learning. We investigated its application to a simulated spoken dialogue system in the flight booking domain, and compare the proposed approach against flat RL. This approach has not been applied before to dialogue systems and the results are promising. Our experimental results confirm those reported by researchers in reinforcement learning - hierarchical learning finds cheaper and faster solutions than flat learning with near-optimal policies. In our case study the hierarchical search space only used 0.64% of the size of the flat search space. Our results report that the learnt policies outperformed a hand-crafted one in three different situations of ASR confidence levels. In addition, our results also report that hierarchical learning converged four orders of magnitude faster than flat learning with a small loss in optimality (on average 0.3 system turns). These results suggest that the proposed approach can be applied to complex and large-scale spoken dialogue systems.

As future work we plan to combine the proposed approach with constrained SMDPs [5], to evaluate the simulation framework, to learn behaviour for a large-scale spoken dialogue system, and to perform tests on an environment with real users.

6. Acknowledgements

Heriberto Cuayáhuitl was sponsored by PROMEP and the Autonomous University of Tlaxcala (http://promep.sep.gob.mx, www.uatx.mx), and Oliver Lemon by ESPRC - EP/E019501/1.

7. References

[1] Levin, E., Pieraccini, R., and Eckert, W. A Stochastic Model of Human Machine Interaction for Learning Dialog Strategies. In *IEEE Trans. on Speech and Audio Processing*, 8:1, 2000.

[2] Scheffler, K. Automatic Design of Spoken Dialogue Systems. PhD Thesis, Cambridge University, 2002.

[3] Pietquin, O. A Framework for Unsupervised Learning of Dialogue Strategies. PhD Thesis, Faculté Polytech. de Mons, 2004.

[4] Williams, J. D. Partially Observable Markov Decision Processes for Spoken Dialogue Management. PhD Thesis, Cambridge University, 2006.

[5] Cuayáhuitl, H., Renals, S., Lemon, O., and Shimodaira, H. Reinforcement Learning of Dialogue Strategies Using Hierarchical Abstract Machines. In *Proc. of IEEE/ACL SLT*, 2006.

[6] James Henderson, Oliver Lemon, and Kalliroi Georgila. Hybrid Reinforcement/Supervised Learning for Dialogue Policies from COMMUNICATOR data. In *Workshop on Knowledge and Reasoning in Practical Dialogue Systems (IJCAI)*, 2005.

[7] Barto, A., and Mahadevan, S. Recent Advances in Hierarchical Reinforcement Learning. In *Discrete Event Dynamic Systems: Theory and Applications*, Kluwer Academic Pub., 13, 2003.

[8] Dietterich, T. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. In *JAIR*, 13, 2000.

[9] Parr, R., and Russell, S. Reinforcement Learning with Hierarchies of Machines. In *Proc. of NIPS*, 1998.

[10] Sutton R., Precup, D., and Singh, S. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. In *Artificial Intelligence*, 112, 1999.

[11] Pineau, J., Roy, N., and Thrun, S. A Hierarchical Approach to POMDP Planning and Execution. In *Workshop on Hierarchy and Memory in Reinforcement Learning (ICML)*, 2001.

[12] Lemon, O., Liu, X., Shapiro, D., and Tollander, C. Hierarchical Reinforcement Learning of Dialogue Policies in a development environment for dialogue systems: REALL-DUDE. In *Workshop on the Semantics and Pragmatics of Dialogue (Brandial)*, 2006.

[13] Dietterich, T. An Overview of MAXQ Hierarchical Reinforcement Learning. In *Proc. of SARA*, 2000.

[14] Watkins, C.J.C.H. Learning from Delayed Rewards. PhD Thesis, King's College, 1989.